
# Self-configured multipath routing using path lifetime for video-streaming services over Ad Hoc networks

Mónica Aguilar Igartua, Víctor Carrascal Frías

PII: S0140-3664(10)00280-X
DOI: 10.1016/j.comcom.2010.06.019
Reference: COMCOM 4310

To appear in: *Computer Communications*

Received Date: 1 August 2009
Revised Date: 11 May 2010
Accepted Date: 11 June 2010

Please cite this article as: n.A. Igartua, c.C. Frías, Self-configured multipath routing using path lifetime for video-streaming services over Ad Hoc networks, *Computer Communications* (2010), doi: 10.1016/j.comcom.2010.06.019

# Self-configured multipath routing using path lifetime for video-streaming services over Ad Hoc networks

Mónica Aguilar Igartua (corresponding author), Víctor Carrascal Frías
{maguilar, vcarrascal}@entel.upc.edu
telephone: +34934015997, fax: +34934011058
Department of Telematic Engineering, Technical University of Catalonia (UPC)
Jordi Girona street 1-3, 08034 Barcelona, Spain

**Abstract**

The increasing spread of mobile nodes along with the technical advances in multi-hop MANETs (Mobile Ad hoc NETworks) make this kind of networks an important type of access network of next generation. The demand of multimedia services from these networks is expected to significantly grow in the next years. Multimedia services, though, require the provision of Quality of Service (QoS). Nevertheless, the highly dynamic nature of MANETs, the energy constraints, the lack on centralized infrastructure and the variable link capacity, makes the QoS provision over MANETs a matter that challenges attention. These features make self-configuration and system adaptation questions of major importance when developing a QoS-aware framework. To tackle this issue, we have designed a-MMDSR (adaptive-Multipath Multimedia Dynamic Source Routing), a multipath routing protocol able to self-configure dynamically depending on the state of the network. The approach includes cross-layer techniques especially designed to improve the end-to-end performance of video-streaming services over IEEE 802.11e Ad Hoc networks. Besides, a straightforward analytical model to estimate the path error probability is presented. This model is used by the routing scheme to estimate the lifetime of the paths. In this way, proper proactive decisions can be made before the paths get broken. The model simplicity is appropriate for low capacity wireless devices. Simulation results validate the proposal and show the improvement on standard DSR (Dynamic Source Routing) and on a previous static version.

*Keywords*: Mobile ad hoc networks, multipath routing, path lifetime, video-streaming services, self-configuration, IEEE 802.11e, cross-layer design.

## 1 Introduction

A Mobile Ad hoc NETwork (MANET) consists of a set of portable computational devices equipped with a wireless communication interface such that nodes are capable of communicating with each other. MANETs have neither

fixed network infrastructure nor administrative support, thus nodes themselves must discover and maintain routes through the network. Since the transmission range of wireless network interfaces is limited, intermediate nodes may be needed, thus each node may operate as a terminal node or as a router to forward packets of other mobile nodes. Nodes are free to move arbitrarily and their batteries have limited capacities, which produce frequent changes in the network topology. Consequently, MANETs must adapt dynamically to be able to maintain on-going communications in spite of these changes [1].

MANETs have attracted a lot of attention from the research community in the last decade and important technical advances have arisen as a consequence. They are envisioned as an important type of access network of next generation, from which multimedia services are surely going to be increasingly demanded by end users. MANETs may be used in a great variety of scenarios, such as universities, museums, emergency rescue or exploration missions, where video-streaming services are likely to be used. These services require the provision of QoS (Quality of Service), which still remains an open issue in MANETs. The special characteristics of MANETs, such as dynamic nature, energy constraints, lack of centralized infrastructure and variable link capacity, make the QoS provision over these networks a challenging objective. These issues make self-configuration and system adaptation features of major importance. In addition, since the QoS provided by a network does not depend on any single network layer but on the different technical specifications of the protocol stack, it is advisable to develop dynamic solutions based on cross-layer approaches able to coordinate the QoS-aware actions of all layers [2].

Our research aims at developing a QoS-aware self-configured adaptive framework suitable to provide video-streaming services over Ad Hoc networks. The framework includes a service-aware multipath routing protocol able to self-configure dynamically depending on the state of the network. Our approach is named a-MMDSR (adaptive-Multipath Multimedia Dynamic Source Routing). It includes cross-layer techniques which improve the end-to-end performance of video-streaming services over IEEE 802.11e Ad Hoc networks. A straightforward analytical model to estimate the path error probability is presented, which is used by source nodes to estimate the lifetime of the available paths to their destinations. This will help source nodes to take proper routing decisions. Simulation results validate the proposal and show the improvement on standard DSR protocol as well as on a previous static version named static-MMDSR (s-MMDSR) [3].

The rest of the paper is organized as follows. Section 2 introduces some related work. The framework is presented in section 3. In section 4 we describe the features of the proposed multipath routing protocol. In section 5, we present an analytical model to dynamically estimate the period of the iterative algorithm that refreshes the multipath scheme. The implementation of the approach in the framework is summarized in section 6. Simulation results are shown and analyzed in section 7, and the analytical model is validated in section 7.1. Finally, concluding remarks and future work are given in section 8.

## 2 Related research

Early MANET routing protocols, by the start of the nineties, focused on finding the shortest path from a source to its destination. Shortly afterwards, proposals considered routing metrics such as link quality, available bandwidth, and signal strength. The analysis on the characteristic of route lifetime in MANETs is quite limited. Some researchers have focused their efforts on developing proposals that try to extend the lifetime of the paths in MANETs. There are mainly two important factors that adversely affect the route lifetime and consequently the network lifetime: node mobility and battery capacity. In mobile ad hoc networks nodes may move away from their neighboring nodes, which causes frequent link failures that invalidate every route containing those links. In addition, battery is a scarce resource and nodes may suffer a lack of energy, so that the network may get disconnected as well. Besides, once a link is detected broken, an alternative route has to be discovered, incurring extra route discovery overhead and increasing delay as well. Thus, routing protocols that take into account residual path lifetime are of major interest.

For the present, we have focused our approach to calculate path lifetime regarding node mobility and leave battery capacity to be tackled in a future research. The reason is that the service under consideration, i.e. video-streaming over MANETs, is more likely to be constrained by node mobility than by battery capacity. To illustrate this, we can consider a group of tourists in the historical center of a city downloading related videos during the tour; they probably will recharge their cellulars at night in the hotel.

One of the first works, widely referenced, that addressed formally the issue of path lifetime regarding mobility is [4], where McDonald and Znati developed a mobility model for ad hoc networks which they used to derive expressions for the probability of path availability as a function of time. In that paper, they used a Rayleigh distribution to represent the distance traversed by a node. Their approach seems to be very simple, with the exception of the possible drawback due to the infinite number of summands in the analytical expression. Alternatively, we are interested in developing closed form expressions easy to compute by the light nodes in MANETs. Obtaining a simple and easy to compute expression is essential, as nodes in MANETs have a limited battery power and therefore their computation capabilities are scarce as well. Even though the price to pay would be a certain loss in accuracy, simplicity is fundamental, as long as the level of precision remains reasonable. The authors in [5] proposed an analytical model for path duration in multi-hop wireless networks in a similar way than [4]. In this case, they used a lognormal function as the distribution of the link duration, which they proved to fit better than other distributions (e.g. exponential or Rayleigh) after carrying out several simulations. Nevertheless, again the drawback of infinite number of terms arises in the computation of the path duration.

In [6], the authors deal with an empirical algorithm to estimate the residual path lifetime of the routes. It consists in a simple mechanism included in each node that collects link lifetime statistics from the beacon messages periodically broadcasted by the nodes. The link lifetime is estimated as the duration between the first and the last time a node hears its neighbor' beacon. Each node continuously collects link lifetime statistics in this fashion until a sufficient number of them has been collected to create a histogram. The simplicity of

the methodology has the limitation of estimating the probability distribution function from actual statistics, which implies waiting to have enough samples before using them with certain reliability.

Nevertheless, to the best of our knowledge, there is no proposal that makes use of the service characteristics to improve the path lifetime of the network. In this paper, we propose a service-aware self-configured multipath-routing framework able to adapt to the inherent dynamics of a MANET. Our routing protocol not only has the ability to find long-living paths, but also it proactively starts to search alternative paths before the actual breakage of the current paths. We have developed a mathematical model to estimate the path error probability of the forwarding routes. The straightforward analysis results in a closed form and easy to compute mathematical expression suitable to be used by light mobile devices to foresee path error probabilities of the available paths. In addition, the multipath scheme is arranged taking the importance of the video-frames into account. Simulation results validate the analytical model to estimate the path lifetime and show the improvement of our proposal.

# 3 Underlying of the framework

Our framework uses the MAC (Medium Access Control) IEEE 802.11e [7], which supports the provision of QoS by including some improvements to the MAC layer. It is suitable for delay and losses sensitive applications such as video-streaming or voice over wireless networks. Our framework has been designed specifically for video-streaming services over MANETs. To achieve this, we have taken the features of the video format into consideration to arrange the four Access Categories (AC) of the IEEE 802.11e MAC level. In this section we give the main details of the type of service to be provided and its integration with the MAC level.

## 3.1 Video codification

Video is distributed using RTP/RTCP (Real Time Protocol/Real Time Control Protocol) over UDP as transport protocols. One of the most used data types in video-streaming is MPEG-2 hierarchical scalable multi-layer encoded video [8]. Layered coding allows enhanced layers of several qualities to be transmitted, given that a minimum bandwidth is guaranteed to transmit a base layer. We use a layered MPEG-2 VBR coding of the video flow. MPEG-2 encoded video is formed by sets of frames, typically somewhere from 4-20 frames each, called GoP (Groups of Pictures). In a GoP there are three types of frames: I, P and B. A GoP follows frame-patterns like IBBPBBPBBPBBPBB. Notice that in a GoP there are one I frame, several P frames and rather more B frames. The size of I, P and B frames are about 4000, 800 and 400 bytes, respectively.

I (Intra) frames encode spatial redundancy. They form the base layer and provide a basic video quality. They carry the most important video information for the decoding process at the receiving side. GoPs could be decoded even if just I frames were present. Besides, I frames are absolutely necessary to decode the video sequence. The entire GoP would be lost if the corresponding I frame was not available at decoding time. P (Predicted) and B (Bi-directional) frames

provide enhancement layers. P and B frames carry differential information from preceding, or preceding and following, I or P frames respectively.

These video characteristics can be taken into account when planning a QoS-aware scheme. For example, different priorities could be assigned to the video frames according to their importance within the video flow. This way, I frames should have the highest priority whereas B frames should have the lowest one.

## 3.2 IEEE 802.11e

The standard defines two different access mechanisms: the Enhanced Distributed Channel Access (EDCA) and the Hybrid Coordination Function Controlled Channel Access (HCCA). The proper access mechanism in MANETs is EDCA, since no centralized access point is needed. The main difference in IEEE 802.11e with respect to the former IEEE 802.11 standard is that there are four different Access Categories (AC), as depicted in Fig. 1. Each packet from the higher layer arrives at the MAC layer with a specific priority value and it is mapped into an AC. Each AC has different parameters in the back-off entity, named *Arbitration Inter-Frame Spacing* (AIFS[AC]), *Minimum Contention Window* ($CW_{min}$[AC]), *Maximum Contention Window* ($CW_{max}$[AC]). Basically, the smaller AIFS, $CW_{min}$ and $CW_{max}$, the shorter the channel access delay, and hence the more capacity share for a given traffic. However, the probability of collisions may go up when operating with smaller $CW_{min}$. In addition, there is another parameter, the *Transmission Opportunity* (TXOP[AC]) defined as an interval of time when a station has the right to initiate transmissions. Finally, each AC has a different *Retry Limit*[AC] value, so that packets are discarded in case the number of retransmissions exceeds that value. These parameters can be used to differentiate the channel access among different priority traffics.

We have defined the mapping of the different packets into each one of the four Access Categories of the IEEE 802.11e MAC as follows:

- AC0: high priority packets (signaling + I frames)
- AC1: medium priority packets (P frames)
- AC2: normal priority packets (B frames)
- AC3: low priority packets (best effort)

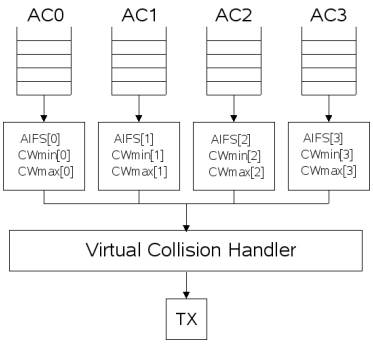


Figure 1: IEEE 802.11e MAC scheme.

# 4 Multipath Multimedia Dynamic Source Routing (MMDSR)

In this section we describe the basic elements of our proposal. We have started from standard DSR [9] as the routing engine to find available paths, since it is suitable to be easily extended for multipath operation. The customer requirements are established by means of a Service Level Agreement (SLA). Such SLA specifies network QoS parameters and their values to provide the committed image quality. The QoS parameters considered are: minimum expected bandwidth ($BW_{min}$), maximum percentage of data losses ($L_{max}$), maximum delay ($D_{max}$) and maximum delay jitter ($J_{max}$):

$$customer\_req \equiv \{BW_{min}, L_{max}, D_{max}, J_{max}\} \quad (1)$$

Notice that in this kind of infrastructureless networks, there is no central unit to enforce SLAs. Instead, we refer to a very simple SLA management, which consists in a simple CAC (Connection Admission Control) to fulfill the user requirements in terms of bandwidth, losses, delay and delay jitter. As it will be seen in section 4.4, this CAC is performed each time new paths are selected to arrange the multipath routing scheme that forwards packets from source to destination. MMDSR has been designed to support multiple video sources. All the decisions (e.g. path selection) and operations (e.g. tuning of configuration parameters) are managed from the source and they depend on the state of the network, so that the framework operation is adaptive to the environment. Let us comment the well-known potential benefits of multipath routing in MANETs, i.e. multiple paths can offer load balancing, better fault-tolerance, and higher aggregate bandwidth, provided that there is a proper algorithm to manage the system seeking an optimal performance.

## 4.1 Multipath scheme

We proved in a previous work [3] that it is not worthwhile to arrange more than three paths simultaneously in a multipath scheme, due to excessive overhead increase and small improvement. Similar works arrived to same conclusion, e.g. [10]. As Fig. 2 depicts, the most important frames of the coded video flows (I frames) are sent through the best available path. P frames are sent through the second best path and B frames through the third. If only two paths were available, I frames would be sent through the best one, and P and B frames through the other one. In case of a unique available path, every frame would be sent together. Let us remark that this distribution of frames over a 3-path scheme contributes to improve the performance of the perceived video quality as simulation results will show.

## 4.2 MMDSR Operation

Prior to the start of a video transmission, a *Probe Message* (PM) packet is sent from source to destination through each one of the $D$ paths discovered by the DSR routing engine. At destination, a time-out is triggered upon the arrival of the first PM packet. Possible PM packets received after expiration are not considered, since they arrived from a path that produces too much delay for

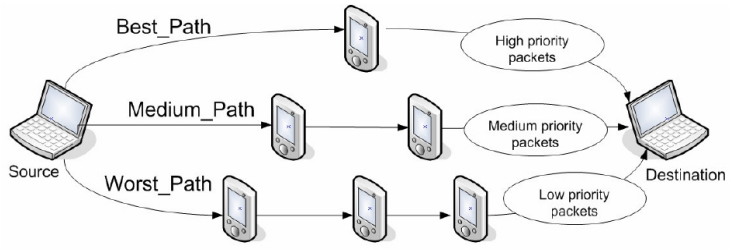


Figure 2: Multipath scheme using three paths.

the service. After time-out expiration, the destination node generates a Probe Message Reply (PMR) packet that contains a set of sampled values of the QoS parameters collected from all the PM packets that arrived in time. The PMR message is sent back to the source through each one of the paths through which a PM arrived, as it is shown in Fig. 3. This information will be analyzed at the source, where a score is assigned to each one of the paths. Then, paths are classified accordingly. After that, the source selects as many paths as needed by the multipath scheme. The QoS parameters computed for each one of the available paths are collected in a vector, named *path-state*:

$$path - state_k^i \equiv \{BW, L, D, J, H, RM, MM\}_k^i \tag{2}$$

where $i$ is the number of iteration of the algorithm and $k$ refers to each one of the $K$ selected paths (with $K \leq D$) to compose the multipath scheme. These parameters are the available bandwidth ($BW_k^i$), the percentage of losses ($L_k^i$), the delay ($D_k^i$), the delay jitter ($J_k^i$), the hop distance ($H_k^i$) and two new QoS parameters we have designed specially for MANETs: *Reliability Metric* ($RM_k^i$) and *Mobility Metric* ($MM_k^i$), which are described below.

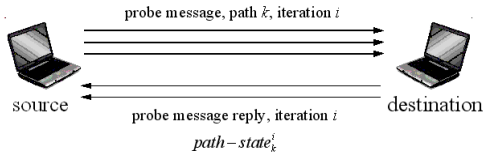


Figure 3: PM and PMR packets.

This process is repeated over time with a certain period (named $T_{routing}$) to refresh the paths. At the beginning of each iteration $i$ of the algorithm the source sends a PM packet through each one of the $D$ discovered paths. In this way, our proposal copes with the dynamic nature of MANETs which may produce frequent link breakages and topology variations through time as a consequence. In a previous static version of our protocol, s-MMDSR (static-MMDSR) [3], this routing period was fixed to 10 sec., as it was shown to be a proper value from numerous simulations performed for typical scenarios. In the current updated version, a-MMDSR (adaptive-MMDSR), the routing period changes dynamically depending on the path error probabilities (PEPs) of the

forwarding routes. Section 5 presents our analytical model to estimate the PEP to assist the multipath routing algorithm with the selection of proper routing periods through time. Simulation results show how this new feature of self-configuring outperforms both plain DSR and the previous static version of the framework.

## 4.3 Computation of QoS parameters

In this section we briefly describe the QoS parameters used to evaluate the quality of the available paths from source to destination. The source uses feedback information to sort the paths regarding their quality and then chooses the best ones to arrange the multipath routing scheme. In this paper we give just the basics of the calculation. The detailed mathematical development was carried out in a previous work [11]. It can also be found in [12]. Basically, the mathematical expressions in [11] consist in a set of inequalities that compare each metric value to certain qualifying thresholds, derived from many representative simulations, and assign qualifications to the path accordingly.

### 4.3.1 *Reliability Metric*, $RM_k^i$

Periodic *Hello Messages* (HM) are interchanged once a second between neighboring nodes to continuously sense the area. We use them to monitor the perceived signal strength. Once a HM is received, the node computes the SINR (Signal-to-Interference plus Noise Ratio) regarding the received packet and attaches this value to a Hello Message Reply (HMR), which is sent back to the source of that HM packet.

We compute the Reliability Metric (RM) as a performance measure of the whole signal quality of a path $k$ for each $i$ iteration of the algorithm. We assign an averaged qualification to each path $k$ computed from the individual SINR values of the links that compose that path. Basically, the higher the SINRs of the links, the higher the RM qualification of the path. For further details we refer the reader to equations (3) to (5) in [11], where straightforward mathematical expressions to compute the RM are presented. Besides, RM qualifications depend on a variable named $NState$ that changes throughout time and tracks the *state of the network* during the previous iteration of the algorithm. That is, the RM for path $k$ during iteration $i$, i.e. $RM_k^i$, depends on $NState^{i-1}$. This way, the framework adapts dynamically to the network variations in such a way that the resolution to mark paths adapts to be able to distinguish paths properly, either under good or bad network performances. Equation (3) shows the calculation of $NState^i$ just averaging all the qualifications in the multipath scheme for the metrics (i.e. bandwidth, losses, delay, jitter, hopcount, RM, MM) obtained from the feedback information carried in the current Probe Message Reply (PMR) packet. The rest of the metrics are defined in the following sections. The weighting values for each parameter, which have to add the unit, will have a higher or lower value depending on the relevance we want to give to each parameter.

$$NState^i = w_{RM} \cdot \overline{RM^i} + w_{MM} \cdot \overline{MM^i} + w_{BWM} \cdot \overline{BWM^i} + \\ + w_{ML} \cdot \overline{ML^i} + w_{MD} \cdot \overline{MD^i} + w_{MJ} \cdot \overline{MJ^i} + w_{MH} \cdot \overline{MH^i} \tag{3}$$

#### 4.3.2 *Mobility Metric*, $MM_k^i$

We use a simple way to infer the degree of relative mobility among nodes without adding overhead. Each node $X$, which belongs to one of the available paths from source to destination, detects the received signal power $RSP_{Y \to X}$ from its neighbors $Y$ from successive periodic *Hello Messages*. Hence, node $X$ computes the relative mobility metric regarding each node $Y$ during iteration $i$, i.e. $M_X^i(Y)$. The $M_X^i(Y)$ values are averaged to obtain $M_X^i$ as expressed in eq. (4), simply from the current and the precedent values of the received signal power. In eq. (4) the mathematical expectation E[.] is estimated using the statistical mean of the observed data.

$$M_X^i = E\left[\left(10log_{10}\frac{RSP_{Y\to X}^i}{RSP_{Y\to X}^{i-1}}\right)^2\right] \tag{4}$$

A low value of $M_X^i$ indicates that node $X$ is relatively less mobile with respect to its neighboring nodes, whereas a high value means that $X$ is highly mobile with respect to its neighbors. A path whose nodes have a lower aggregate relative mobility will be preferred instead of other paths whose nodes have a higher mobility. Then, next time a *Probe Message* arrives at node $j$ within path $k$, the node will append the current value of its mobility metric qualification, $MM_{k,j}^i$, computed as a simple function of (4), which is described in eq. (7) of [11]. Basically, the higher the relative mobility of a node with respect to its neighbors, the lower the assigned mobility metric $MM_k^i$ of path $k$ during iteration $i$. Again, this function depends on the state of the network during the previous iteration, i.e. $NState^{i-1}$, to track the dynamism of the network.

#### 4.3.3 *Hop Metric*, $HM_k^i$

We obtain a measure of the metric of the length of the available paths. It is computed from the number of hops of the longest available path ($h_{max}^i$), and the number of hops of the shortest available path ($h_{min}^i$) in the current algorithm iteration $i$. Qualifications regarding the lengths of the paths are assigned so that shorter paths get higher scores. In general, shorter paths are preferred, as fewer losses will take place due to the contention for the medium produced in every hop. See eq. (9) in [11] for the details of the mathematical assignation.

#### 4.3.4 Losses $LM_k^i$, delay $DM_k^i$ and jitter delay $JM_k^i$ metrics

As soon as the PMR packet arrives at the source node, this node gets the values for losses, delay and delay jitter which have been sampled in each path $k$ in the current iteration $i$. Afterwards, qualifications are assigned in such a way that paths with lower losses, delay and jitter get higher marks for the respective metric. The assignation is fully described in eq. (10) of [11]. Also, those marks depend on $NState^{i-1}$ as in the previous metrics.

#### 4.3.5 End-to-end Bandwidth Metric, $BWM_k^i$

The PM packet also collects $BW_k^i$, which is the end-to-end available bandwidth for each one of the $k$ paths. It corresponds to the bandwidth of the bottleneck link (i.e. the link with lower bandwidth) within the path. $BWM_k^i$ qualifications are applied to each path so that paths with higher available bandwidth get higher scores (eq. (11) in [11]). In the computation of $BWM_k^i$ there is also a dynamic function that reflects the changes in the network state, which (3) tracks.

### 4.4 Path classification

Once the available paths whose qualities collected in eq. (2) fulfill the user's requirements set in eq. (1) have been selected, the algorithm arranges them sequentially according to the next list of qualifying parameters.

1. $RM_k^i + MM_k^i$
2. $HM_k^i$
3. $BWM_k^i$
4. $LM_k^i + JM_k^i$
5. $DM_k^i$

After carrying out a considerable number of simulations in order to decide the most important metrics for the service under evaluation, we have chosen RM and MM as the most important parameters to classify paths, because most reliable and stable paths are preferred to distribute video-streaming services over MANETs, which are inherently so dynamic. In case of draw, the hopcount is the next metric to classify paths since each additional link in a path may increment the chance of collision, so shorter paths are preferred. Delay, jitter delay and losses are lesser determinant metrics for this kind of service over this type of networks, although they are taken into account as well. Delay is the least determinant metric since receiver nodes have a buffer to store video frames waiting to be decoded, so a certain delay can be assumed. Let us remark that the algorithm selects only available paths which already accomplish the user requirements. After ordering the available paths, the source selects the required number of paths to arrange the multipath routing scheme.

## 5 Path Error Probability model

In this section we derive an analytical model to calculate the *Path Error Probability* (PEP) of each forwarding path in the multipath routing scheme. To achieve this, we first model the movement of the nodes in a MANET and then compute the *Link Error Probability* (LEP).

A path is composed by a series of links or hops. Our goal is to find a straightforward expression to analytically estimate the PEP so that source nodes, which are light mobile devices, can use this approach to foresee the path error probabilities of a set of available paths and make proper routing decisions in selecting the best path (or set of paths) to route packets. In addition, source nodes can use this model to switch to a new path (or set of paths) in the proper moment before an actual link breakage takes place in the current set of paths. This means to decide the proper routing period ($T_{routing}$) of the algorithm according to the

evolution of the path error rates. This value will be estimated in the setting moment of the multipath scheme, so that this feature is a proactive action to refresh the multipath routing scheme. The multipath scheme is updated when there is a high probability of path breakage. This computation will be done in such a way that the higher the path error probability, the lower the routing period. This way, under high mobility situations new paths are searched sooner as the topology varies frequently. On the other hand, lower overhead will be produced under stable and favorable situations since paths will be refreshed by new ones later. This new proposal leads to improve the global performance by means of a decrease of the packet losses, an increase of the perceived video quality and a reduction of the routing overhead.

### 5.1 Mobility model in a MANET

For the sake of simplicity, some assumptions must be done to obtain an analitical model. We consider that the movement area is a two-dimensional square area of $W$ meters per side, as depicted in Fig. 4 (a). We assume that nodes move based on a *Random Waypoint Mobility Model* (RWMM) through a homogeneous environment, i.e. all the nodes use the same set of mobility parameters. We summarize the basics of RWMM as follows. The movement of a node is modeled by a sequence of intervals (also named *epochs*) of random length which are i.i.d. (identically, independently distributed) exponentially with average $1/\lambda$ sec. During each interval $i$ of duration $t_i$ sec, a node moves at a constant speed $v_i$ m/s in a straight line at an angle of $\theta_i$ along $l_i$ meters. This distance $l_i$ equals $v_i \cdot t_i$. The direction, $\theta_i$, of the mobile during each epoch is i.i.d. uniformly distributed over (0, $2\pi$) and remains constant only for the duration of the epoch. A scheme of this RWMM is sketched in Fig. 4 (b). Notice that $1/\lambda_i = t_i = l_i/v_i$.

At the end of each interval the node stops during $t_{pause}$ sec. Afterwards, the node starts another interval with other values of speed and direction. We consider $t_{pause}$ uniformly distributed within $[0, t_{pausemax}]$. We have considered pause-time random models in our analytical model given that the speed of the nodes can follow any distribution. We just require to characterize this distribution by means of an average and a standard deviation. It is worth noting that in a general MANET there will be many different nodes with different mobility characteristics. However, we need to make some general assumptions to carry out a simple analytical approach to easily estimate the proper $T_{routing}$, despite a certain loss of accuracy.

Speeds are randomly distributed within an interval $[v_{min}, v_{max}]$ according to a probability distribution which we assume to be Gaussian. Thus, the mobility profile of a node which moves according to a RWMM can be specified with three parameters $(\lambda, \mu_v, \sigma_v)$, being $\mu_v$ and $\sigma_v$ the average and the standard deviation of the speeds of the nodes respectively. Next we obtain the probability distribution function and the cumulative distribution function of the distance moved by a node, to finally obtain LEP and PEP expressions.

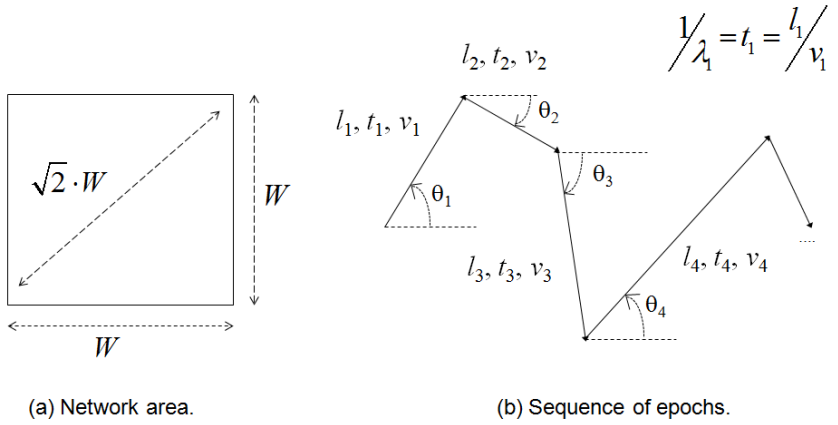


Figure 4: RWMM (Random Waypoint Mobility Model).

## 5.2 Probability distribution function of the distance between two nodes

The probability distribution function (*pdf*) of the distance $r$ between two nodes, $pdf_{2n}(r)$, can be calculated using equation (5). Let us highlight that $r$ is a two-dimensional distance defined in the network area (see Fig. 4). The complete analytical derivation of this expression can be found in [13, 14], where the probability distribution of the distances between two randomly positioned mobile nodes that follow a random waypoint model in a wireless network over a rectangular area is carried out. In both works, they derive the expression applying straightforward geometrical and probabilistic analysis. Section 5.5 uses this *pdf* to compute the PEP.

$$pdf_{2n}(r) = \begin{cases} \frac{2r}{W^2}\left(\pi - \frac{4r}{W} + \frac{r^2}{W^2}\right) & 0 \leq r \leq W \\ \frac{2r}{W^2} 2 \cdot arcsen\frac{W}{r} - 2 \cdot arccos\frac{W}{r} + \\ \quad \frac{4\sqrt{r^2-W^2}-W}{W} - \frac{W^2+r^2}{W^2} & W < r \leq \sqrt{2} \cdot W \\ 0 & \sqrt{2} \cdot W < r \end{cases} \quad (5)$$

## 5.3 Cumulative distribution function of the distance

The magnitude of the epoch length in a RWMM can be approximated by a normal distribution, as it was shown in [5]. Thus, to compute the cumulative distribution function (*CDF*) of the distance $r$ moved by a node we just need the average $\mu$ and standard deviation $\sigma$ of the simple Gaussian distribution. To obtain $\mu$ and $\sigma$, we have used the fact that in a normal distribution the 99% of the values are included in the interval $[\mu - 2,5758 \cdot \sigma, \mu + 2,5758 \cdot \sigma]$. We have equalled this interval to the interval $[0, R]$ of possible distances moved by the node within the cell boundary of transmission range $R$, obtaining $\mu = R/2$ and $\sigma = R/(2 \cdot 2,5758)$. Hence, we propose to approach the *CDF* of the distance $r$ moved by a node by equation (6).

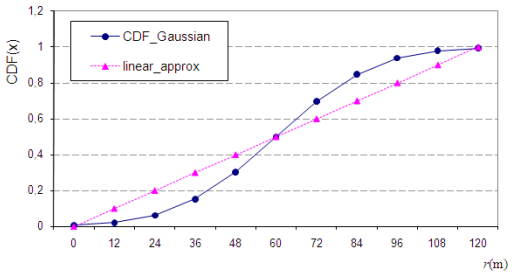


Figure 5: CDF of the distance moved by a node *vs.* its linear approximation ($R = 120$m, $\mu = \frac{R}{2}$ , $\sigma = \frac{R}{2 \cdot 2{,}5758}$).

$$CDF_d(r) = p(d \leqslant r) \cong \frac{1}{2}\left(1 + erf\left(\frac{r-\mu}{\sqrt{2}\sigma}\right)\right) = \tag{6}$$
$$= \frac{1}{2}\left(1 + \frac{2}{\sqrt{\pi}}\sum_{n=0}^{\infty}\frac{(-1)^n\left(\frac{r-\mu}{\sqrt{2}\sigma}\right)^{2n+1}}{n!(2n+1)}\right),\ 0 \leq r \leq R$$

In spite of its apparent simplicity, the drawback of the *erf* function is its infinite number of terms. Thereby, we look for similar functions easier to cope with, seeking for a closed form expression of the CDF of the distance moved by a node. We use a simple distribution function to approach equation (6). Simulation results will show its worthiness after evaluating the trade-off between accuracy and simplicity. Fig. 5 depicts equation (6) ($CDF_{Gaussian}(r)$, line of circles) and its linear approximation ($linear\_approx(r)$, line of triangles) expressed in equation (7).

$$CDF_d(r) = p(d \leqslant r) \cong linear\_approx(r, R) = \frac{r}{R}, 0 \leq r \leq R \tag{7}$$

## 5.4 Estimation of Link Error Probability (LEP)

The *Link Error Probability* (LEP) can be computed using expression (9), where $d$ stands for the random variable of the distance moved by a node; $p(d \leq r)$ is the cumulative distribution function of the distance moved by the node, which has been estimated with (7).

The analysis to obtain (8) was carried out in [15]. Let us consider two nodes which form a link. In (8), $pdf_r(r)$ is the probability distribution function of the distance node B must move to reach the boundary of the cell defined by the reference node A, whose transmission range is $R$. According to the RWMM, node B is initially located anywhere within the cell of A with equal probability, and it moves in a random uniform direction over $(0, 2\pi)$.The probability of link breakage is thus $p(d > r)$, being $d$ the distance moved by node B and $r$ the distance from that node to the cell boundary of A. We obtain a measure of the LEP by integrating the product of $p(d > r)$ with the *pdf* of the distance

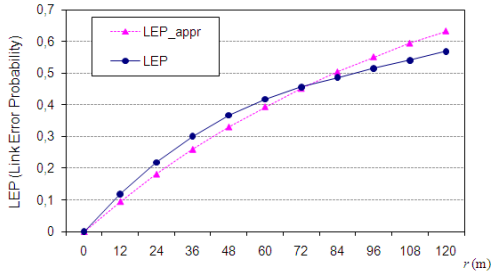


Figure 6: LEP and its exponential approximation ($r = 120$m).

to the cell boundary (i.e. $pdf_r(r)$), over all possible distances between the two nodes. An equivalent procedure is followed in [4]. It is worth remarking that our $CDF_d(r)$ is a simple and closed form expression which produces a simple expression to estimate the LEP.

$$pdf_r(r) = \frac{4}{\pi \cdot R^2} \cdot \sqrt{R^2 - r^2}, 0 \leq r \leq R \tag{8}$$

$$LEP(r,R) = \int_0^r p(d > r) \cdot pdf_r(r) \cdot dr = \int_0^r (1 - p(d \leq r)) \cdot pdf_r(r) \cdot dr = \\ = \int_0^r (1 - CDF_d(r)) \cdot pdf_r(r) \cdot dr, 0 \leq r \leq R \tag{9}$$

The resolution of this integral yields to this approach for the *Link Error Probability*:

$$LEP(r,R) = \int_0^r (1 - CDF_d(r)) \cdot pdf_r(r) dr \approx \int_0^r \left(1 - \frac{r}{R}\right) \frac{4}{\pi R^2} \sqrt{R^2 - r^2} \cdot dr = \\ = \frac{4}{\pi \cdot R^2} \cdot \left[ r \cdot R - 0.22 \cdot r^2 - \frac{R^2}{3} + \frac{(R^2 - r^2)^{3/2}}{3 \cdot R} \right], 0 \leq r \leq R \tag{10}$$

Figure 6 draws the evolution of equation (10) in the circle-shaped line. This function clearly has an exponential behavior. Thus, for the sake of simplicity we can approximate LEP($r$, $R$) by this even simpler expression (triangle-shaped line in Fig. 6), being the average relative error of 9,5%:

$$LEP(r,R) \approx 1 - e^{-\frac{r}{R}}, 0 \leq r \leq R \tag{11}$$

Alternatively, we can express LEP as a function of the transmission range $r$, the average speed of the nodes $\mu$ and the time $t$.

$$LEP(R,\mu,t) \approx 1 - e^{-\frac{\mu \cdot t}{R}}, 0 \leq t \leq \frac{R}{\mu} \tag{12}$$

All in all, this seems to be a proper analytic model to be computed by the light mobile nodes in a MANET, due to a good trade-off between simplicity and accuracy. The expression of the *Link Error Probability* (LEP) will allow us to easily estimate the *Path Error Probability* (PEP).

### 5.5 Estimation of Path Error Probability (PEP)

Once an approach for the LEP has been obtained, we can easily extend it to characterize the probability that an existing path breaks, i.e. the *Path Error Probability* (PEP). We relate PEP to individual LEPs according to the assumptions of same LEP for all the links in the path and independent link failures. We average each possible PEP value for a path with a given number of hops, weighted by the probability of having that number of hops in the path. We consider all the possible number of links (hops) that a path can have. In (13), the LEP is estimated using equation (12).

$$PEP(N, W, R, \mu, t) = 1 - \sum_{i=1}^{H} (1 - LEP(R, \mu, t))^i \cdot p(i\ hops) \tag{13}$$

In equation (13), $H$ is the maximum number of hops in a path in the squared area of the network (see Fig. 4 a)). $H$ is expressed in (14), where $N$ is the number of nodes, $W$ is the size of the squared network and $R$ is the transmission range of the nodes. In this case, the maximum number of hops would be achieved when nodes were arranged in the hypotenuse of the square area, so the number of hops would be $\sqrt{2} \cdot W/R$. However, obviously the maximum number of hops can not exceed $N$-1 hops. In this expression, we assume there are enough nodes in the network so that the shortest forwarding path between source and destination is the Euclidian distance. This hypothesis leads to this simple relation for the maximum number of hops in any path:

$$H = min\left(N - 1, \lfloor \frac{\sqrt{2} \cdot W}{R} \rfloor\right) \tag{14}$$

The probability that there are exactly $i$ hops between two nodes, $p(i\ hops)$, can be computed with equation (15), where $pdf_{2n}(r)$ is the probability distribution function of the distance between two nodes which was expressed in equation (5). To calculate equation (15) it can be seen that it is necessary to compute two indefinite integrals which are solved in equations (16) and (17).

$$p(i\ hops) = \int_{(i-1)\cdot R}^{i \cdot R} pdf_{2n}(r) \cdot dr \tag{15}$$

$$\int pdf_{2n,0\leq d\leq W}(r) \cdot dr = \int \frac{2r}{W^2} \cdot \left(\pi - \frac{4r}{W} + \frac{r^2}{W^2}\right) \cdot dr = \frac{\pi}{W^2} r^2 - \frac{8}{3W^3} r^3 + \frac{1}{2W^4} r^4 \tag{16}$$

$$\begin{aligned} \int pdf_{2n,W\leq d\leq \sqrt{2}W}(r) \cdot dr = \int \frac{2r}{W^2} \cdot (2 \cdot arcsen\left(\frac{W}{r}\right) - 2 \cdot arccos\frac{W}{r} + \\ + \frac{4 \cdot \sqrt{r^2 - W^2} - W}{W} - \frac{W^2 + r^2}{W^2}) \cdot dr = \frac{2r^2}{W^2} \cdot \left(arcsen\left(\frac{W}{r}\right) - arccos\left(\frac{W}{r}\right)\right) + \\ + \frac{4r}{W}\sqrt{1 - \left(\frac{W}{r}\right)^2} - \frac{(W^2 - rx^2)\left(8\sqrt{r^2 - W^2} - 3W\right)}{3W^3} - \frac{r^4 + 2 \cdot W^2 r^2}{2W^4} \end{aligned} \tag{17}$$

Finally, using equations (14) and (15) in equation (13), we obtain an expression to estimate the *Path Error Probability* (PEP).

To sum up, once we have set the number of nodes in the network ($N$), the dimension of the squared network ($W$), the transmission range of the nodes ($R$) and the average speed of the nodes ($\mu$), equation (13) gives an estimation of the PEP over time. This expression allows source nodes to estimate the proper period of time, $T_{routing}$, in which the routing algorithm should look for an alternative forwarding path. This would happen when the PEP of the current path being used is higher than a given threshold, which we name $P_{threshold}$. This way, the routing protocol uses equation (18) to compute the $T_{routing}$ for every new path established to forward data from source to destination. Once this timeout triggers, the routing protocol will start looking for an alternative path. Let us highlight the proactive feature of our proposal, since paths are refreshed when they are near failure but prior to actual breakage. This will definitively improve the service performance. Notice that without including this proactive proposal, paths would be replaced by others only after breakages, which are frequent in MANETs, producing unpleasant damages in the service performance. Section 7.1 refers to the accuracy of this model compared to simulation results.

$$PEP(N, W, R, \mu, T_{routing}) = P_{threshold} \tag{18}$$

Equation (18) is a nonlinear equation which can easily be solved e.g. applying the simple bisection method [16], which iteratively finds the solution. In all our cases there have just been necessary up to 6 iterations for a relative error of 0.5 sec., which is accurate enough to obtain the proper moment $T_{routing}$ to start finding a new forwarding route. Notice that $T_{routing}$ depends on the same parameters as PEP, i.e. the number of nodes ($N$), the network size ($W$), the transmission range ($R$), the nodes speed ($\mu$) and the time ($t$). That is, $T_{routing}(N, W, R, \mu, t)$.

Notice that our PEP approach adapts to the network features since it depends on the scenario settings ($N$, $W$, $R$, $\mu$). In real life these values should be estimated by the source nodes, although more exactly we could define a set of usual values for typical scenarios, e.g. (20 nodes, 200 m, 50 m, 0-1 m/s) in a museum, (50 nodes, 500 m, 120 m, 1-2 m/s) in a campus or (100 nodes, 1000 m, 120 m, 5-10 m/s) in a city. Once we have developed an approach to estimate the PEP, we are ready to use it to assist the routing protocol in finding the proper moment to switch to an alternative path. This happens when PEP exceeds a given threshold. These moments will determine the working period of the routing algorithm. To illustrate the approach, we analyze in the appendix a numerical example to compute the routing period to refresh paths.

# 6 a-MMDSR with the PEP estimation

Let us summarize how the framework will apply the analytical model to estimate the PEP and how source nodes are going to use the proposal to manage their multipath routing schemes. After reception of a *Probe Message Reply* (PMR), the source proceeds to compute a measure of the path error probabilities of each one of the $k_{th}$ available paths, named $PEP(k)$. After this, each $PEP(k)$ is compared to the established $P_{threshold}$, so that only paths with lower values of

their PEP are valid to be used by the a-MMDSR routing algorithm to establish a multipath routing scheme.

The multipath routing scheme in a-MMDSR arranges a set of paths to forward the I, P and B frames of the video flow according to their qualities, so that the most important frames are sent through the best path. Each new $k_{th}$ path has associated a life time period $T_{routing}(k)$, which is computed from our analytical approach of the Path Error Probability (PEP). Basically, $T_{routing}(k)$ is the time in which the PEP of path $k$ exceeds the given threshold, $P_{threshold}$. A unipath routing scheme would decide to switch the current forwarding path to an alternative path after $T_{routing}$ sec.

Nevertheless, we still have to decide the proper $T_{routing}$ for a multipath routing scheme, since it is composed by a set of paths $k$ each one with its own PEP($k$) and therefore with its own $T_{routing}(k)$. Each $k_{th}$ path has associated a $T_{routing}(k)$ moment computed by the source node in its establishment moment. Therefore, we have to determine a single $T_{routing}$ among a set of $T_{routing}(k)$ values. Among several options, we have chosen a conservative option setting $T_{routing}$ to the $T_{routing}(k)$ value of the $k_{th}$ path which carries the most important video frames (i.e. I frames). This way, the set of paths are refreshed once the best path exceeds the $P_{threshold}$. Other alternatives to compute $T_{routing}$ would be for instance to average all the $T_{routing}(k)$ values or to set the $T_{routing}(k)$ to the minimum $T_{path\_life}(k)$. This latter option would produce a larger overhead, since paths would be refreshed sooner. Our option seeks to use the best path longer although some worse paths may have broken down. This option has shown a good trade-off between performance and routing overhead. It gives enough time to the routing scheme to request and find alternative paths before the current ones actually break. Notice that adding this feature, $T_{routing}$ depends on the quality of the path that carries I frames, which makes the proposal be service-aware.

# 7 Simulation results

We have included the proposal in our framework implemented with the open source network simulator ns-2 (v2.27) [17] Section 7.1 validates the proposal with simulation results, whereas section 7.3 carries out a performance evaluation of the proposal.

## 7.1 Validation of the analytical model

In this section we validate the analytical tool we have developed to estimate the PEP. To achieve this, we compare analytical to simulation results. We consider the controlled scenario depicted in Fig. 7, where five nodes are moving away from each other at a certain speed $v_i$, where $v_4 > v_3 > v_2 > v_1 > v_0$. Several average speeds of the nodes have been considered from 1 to 5 m/s. This represents a worst case in which nodes of a path move away from each other, so the probabilities of having a link broken over time are higher as time goes by.

Fig. 8 shows the evolution of the PEP throughout time. In this case the network parameters are: $N$=5 nodes, $W$=500m, $R$=120m. As expected, the higher the speed of the nodes, the sooner the path would reach a higher probability of breakage. The maximum value of PEP after which the source node should find


Figure 7: Scenario to validate the analytical model.

an alternative path is set to 60%, i.e. $P_{threshold} = 0.6$. As it has been done in the numerical example, we derive the $T_{routing}$ moment from equation (18) just setting $t$=$T_{routing}$ when PEP equals $P_{threshold}$.

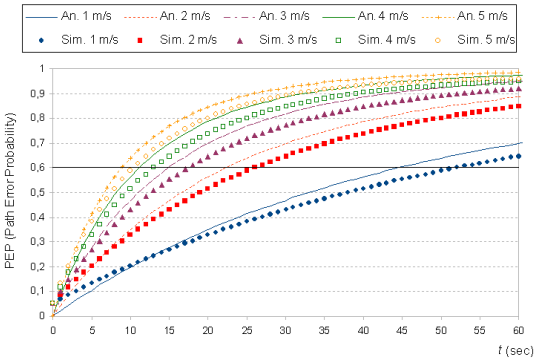


Figure 8: PEP, analysis *vs.* simulation ($R$=120m, $W$=500m, $H$=4 hops).

Table 1 shows $T_{routing}$ values for several average speeds of the nodes. Fig. 8 shows a good accordance between analytical (solid lines) and simulation (dotted lines) results. We can see that our model is slightly pessimistic, since it offers a higher PEP than the values obtained from the simulations. It means that the analysis produces an earlier $T_{routing}$ compared to simulation results, as Table 1 shows. This behavior is suitable, since a more conservative scheme will protect further the performance of the service.

Table 1: Routing period of the algorithm, analysis *vs.* simulation.

| $\mu$ (m/s) | Analytical $T_{routing}$ (s) | Simulation $T_{routing}$ (s) |
|---|---|---|
| 1 | 43.77 | 53 |
| 2 | 21.85 | 25.98 |
| 3 | 14.58 | 16.66 |
| 4 | 10.96 | 12.8 |
| 5 | 8.79 | 10.5 |

Finally, we have carried out a set of simulations in a general MANET scenario whose simulation settings are depicted in Table 3, except here the number of nodes $N$ and the transmision range $R$ vary. Table 2 shows $T_{routing}$ values for different speeds of the nodes. In the first two columns $N$=100 nodes, $R$=75m; for the two next columns $N$=100 nodes, $R$=150m; and $N$=200 nodes, $R$=75m in the last two columns. Again, the analysis produces an earlier $T_{routing}$ compared

to simulation results, which protects further the performance. We can observe how routes last longer for a higher transmission range $R$, since links remain longer. Also, increasing $N$ makes $T_{routing}$ slightly rise, since nodes are closer.

Table 2: $T_{routing}$ (s) varying $N$, $R$. Analysis (A) *vs.* simulation (S).

| $\mu$ (m/s) | $T_A(N100, R75)$ | $T_S(N100, R75)$ | $T_A(N100, R150)$ | $T_S(N100, R150)$ | $T_A(N200, R75)$ | $T_S(N200, R75)$ |
|---|---|---|---|---|---|---|
| 0.5 | 14 | 15 | 120 | 135 | 14.2 | 16 |
| 0.8 | 8.2 | 9.5 | 100 | 105 | 8 | 8.2 |
| 1 | 5.8 | 7 | 77 | 79 | 6.2 | 7.8 |
| 1.5 | 3.9 | 4.5 | 53 | 55 | 4.2 | 5.8 |
| 2 | 3.2 | 3.8 | 40 | 42 | 3.3 | 3.6 |

## 7.2 Performance evaluation

In this section we carry out a performance evaluation of our proposal a-MMDSR compared to the static version s-MMDSR [3] and to plain DSR [9]. Simulation settings are shown in Table 3. Results from other scenarios can be consulted at [12]. A 300 sec. sequence of an MPEG-2 VBR codification of the movie *Blade Runner* is sent over a 500x500 $m^2$ MANET composed of 50 mobile nodes with a 120 m transmission range. Nodes move at speeds in the range 1m/s to 5m/s. We have chosen a typical campus scenario where several tens of users equipped with wireless devices wander around the campus, basically on foot or bicycle. We have taken the sizes of our campus (Campus Nord of UPC, Barcelona, Spain) to give an example.

The MAC IEEE 802.11e access parameters used in the simulation are shown in Table 4. TXOP equal to 0 indicates that only one packet is transmitted per interval. I frames are sent through the highest priority AC[0], P frames through AC[1], B frames through AC[2] and interfering traffic uses the lowest priority AC[3].

To obtain reliable results, we have generated 10 different RWMM scenarios for each one of the speeds. These scenarios have been generated with the Bonnmotion tool [18]. We have run simulations 3600 sec. longer than required, then we have eliminated the first 3600 sec. in order to minimize the impact of the initial transient of the mobility pattern. One transmission of video is present at each simulation between two nodes, and the frame losses of this video are measured. An interfering CBR traffic of 3 Mbps is present between source and destination, in order to simulate a congested network.

Fig. 9 shows the total percentage of packet losses for each one of the routing protocols under evaluation, i.e. DSR, s-MMDSR and a-MMDSR. Confidence intervals of 80% are shown, obtained after carrying 10 simulations per value. It can be seen that both a-MMDSR and s-MMDSR protocols outperform the standard DSR protocol, thanks to their capability to adapt to the environment. The benefits are mainly due to the fact that they use multipath techniques along with a cross-layer algorithm which collects feedback information of the network state. Sources use this feedback information to arrange data through the most proper paths depending on the importance of the video frames within the video flow. Furthermore, a-MMDSR shows a better performance than the static version s-MMDSR because it is able to adjust its configuration parameters dynamically depending on the network state, thus the routing overhead

Table 3: Simulation settings.

| | |
|---|---|
| Area | 500x500 m |
| Number of nodes | $N$=50 |
| Medium nodes speed | $\mu$=1, 2, 3, 4, 5 m/s |
| Transmission range | $R$=120 m |
| Mobility pattern | Random Waypoint |
| MAC specification | IEEE 802.11e |
| Nominal bandwidth | 11 Mbps |
| Simulation time | 300 sec. |
| Video coding | MPEG-2 VBR |
| Video bit rate | 150 Kbps |
| Video sequence | Blade Runner |
| Transport protocol | RTP/RTCP/UDP |
| Maximum packet size | 1500 bytes |
| Multipath scheme | $K$=3 paths |
| Weighting values (eq. (3)) | 1/7 |
| Queue sizes | 50 packets |
| Interfering CBR traffic | 3 Mbps |
| Channel noise | -92 dBm |
| $PEP_{threshold}$ | 0.6 |

Table 4: IEEE 802.11e access categories.

| Parameter | AC[0] | AC[1] | AC[2] | AC[3] |
|---|---|---|---|---|
| AIFS | 2 | 2 | 3 | 7 |
| $CW_{min}$ | 7 | 15 | 31 | 31 |
| $CW_{max}$ | 15 | 31 | 1023 | 1023 |
| TXOP (msec) | 3 | 6 | 0 | 0 |

decreases and the resources are more efficiently used.

In Fig. 10, I-P-B frame losses are shown for the new dynamic framework a-MMDSR compared to the previous static s-MMDSR and to plain DSR. As we can see, the most important frames, i.e. I frames, are protected in both s-MMDSR and a-MMDSR, as they are sending I frames through the best available path using high priority access category (i.e. AC[0]). I frame losses with s-MMDSR (triangle-shaped dashed line at the bottom) are very small, except for high speeds (5 m/s) reaching 10%. I frame losses with a-MMDSR (triangle-shaped solid line at the bottom) are even smaller, showing the same behavior. P frame losses are represented in the second pair of lines from the bottom (circle-shaped dashed line for s-MMDSR and circle-shaped solid line for a-MMDSR). The third pair of lines from the bottom correspond to B frame losses (squared dashed line for s-MMDSR and squared solid line for a-MMDSR). We can see how s-MMDSR and a-MMDSR treat frames with different priorities according to their importance within the video sequence. Frame losses with a-MMDSR are slightly lower than with s-MMDSR, thus the dynamic scheme improves further the system performance. On the contrary, DSR produces almost the

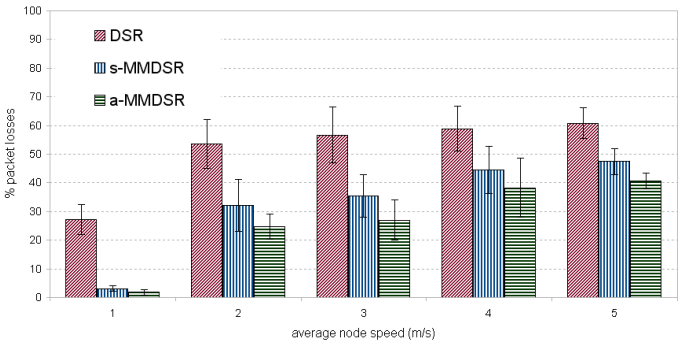


Figure 9: Percentage of packet losses *vs.* nodes' speed.

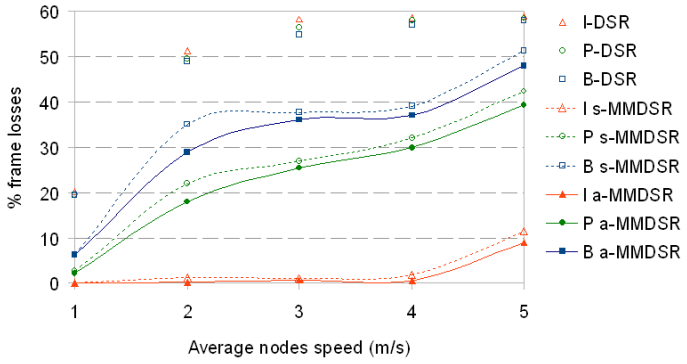


Figure 10: Percentage of I, P, B frame losses *vs.* nodes' speed.

same percentage of losses for the three type of video frames, represented in the three dotted lines at the top of the figure. The reason for this is that DSR does not differentiate video frames according to their importance.

Fig. 11 shows how the percentage of frame losses evolve through time for each one of the protocols in a simulation 300 sec. long. We can observe how both a-MMDSR and s-MMDSR protocols protect important frames (i.e. I frames, whose losses are almost null). In addition, slightly lower losses are achieved with a-MMDSR than with s-MMDSR. At the end of the simulation both a-MMDSR and s-MMDSR protocols offer 1-2% cumulative losses for I frames, around 15% losses for P frames and around 25% cumulative losses for B frames. On the other hand, DSR produces cumulative losses above 35% to whatever kind of frame.

Figure 12 shows the profit of our proposal compared to plain DSR in terms of the PSNR (Peak Signal to Noise Ratio), which is an objective measurement

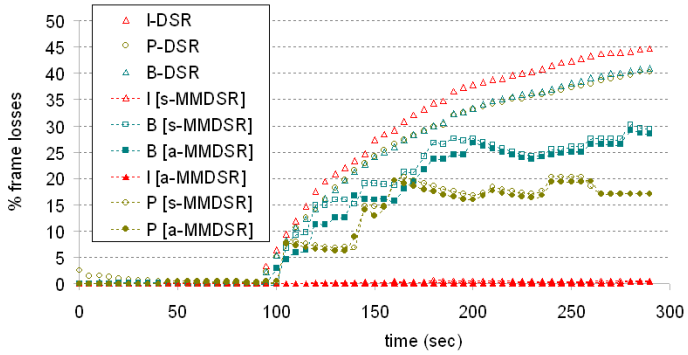


Figure 11: Percentage of I, P, B frame losses over time.

of the video quality experienced by the users. Five simulations have been carried out and confidence intervals of 99% are shown. In order to evaluate the importance of this improvement, let us highlight some numbers about the relation between the PSNR and the users' Mean Opinion Score (MOS), which provides a subjective evaluation of the video quality experienced by the users. According to ITU-T recommendation P.801 [19], the MOS evaluation can be *Excellent* (PSNR above 30dB), *Good* (PSNR about 29dB), *Fair* (PSNR about 28dB), *Poor* (PSNR about 26dB) or *Bad* (PSNR below 25dB). This relationship between MOS and PSNR can be found in the specific video literature, e.g. [20], which are based on user polls. Hence, according to Fig. 12 the service performance with DSR would be *Bad* whereas with a-MMDSR would be *Excellent*.

In Fig. 13 we can see the evolution of the routing period in a-MMDSR for a simulation 900 sec. long. We can see that the fixed routing period of s-MMDSR (triangle-shaped line) is set to $T = 10$ sec. This means that s-MMDSR refreshes the routes every 10 secs, independently of the fact of being the paths still useful or not. Let us recall that with our proposal $T_{routing}$ adapts dynamically depending on the network scenario settings and on the time, i.e. $T_{routing}(N, W, R, \mu, t)$. This feature avoids deciding a proper $T_{routing}$ moment and lets the framework compute it instead. We can see that 81% of the time $T_{routing} > T$ for a-MMDSR (circle-shaped line), so paths are refreshed later than with s-MMDSR. This feature produces a decrement in the overall signaling overhead, as the scheme looks for new routes less often while the current ones are still reliable. The traffic due to overhead is lower in a-MMDSR than in s-MMDSR, thanks to the dynamic routing period and adaptive configuration. In average, there is a reduction of 20% of the signaling overhead in a-MMDSR compared to s-MMDSR, as Table 5 shows. This fact produces a better use of the resources, which are so scarce in MANETs. The overhead traffic in MMDSR includes standard DSR control packets and MMDSR monitorization packets. Although in both MMDSR cases the overhead is higher compared to standard DSR, it is worthwhile to include the multipath QoS-aware routing scheme since

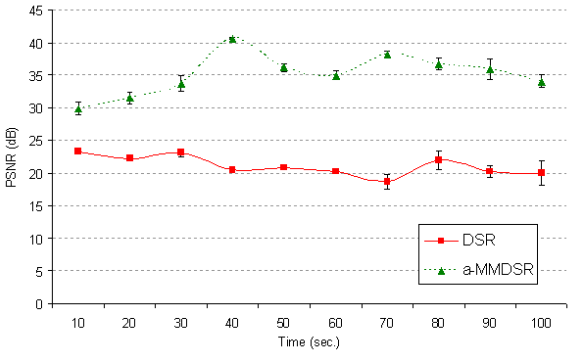


Figure 12: PSNR over time for DSR and a-MMDSR.

the performance is better, as it has been shown in this section.

Table 5: Signaling overhead.

| Routing protocol | %overhead |
|---|---|
| DSR | 0.4% |
| s-MMDSR | 6.5% |
| a-MMDSR | 5.2% |

### 7.3 Performance under other mobility models

We have carried out the validation of our proposal and a performance evaluation under Random Waypoint (RW) mobility model. Now, we will see the operation under more realistic patterns, such as RPGM (Reference Point Group Mobility) and MG (Manhattan Grid), which can be briefly described as follows.

- Reference Point Group Mobility (RPGM): Each group has a logical centre (group leader) that determines the group's motion behavior. E.g.: Group of tourists following a leader in downtown, or rescue crews that work cooperatively during disaster relief.
- Manhattan Grid (MG): It emulates the movement pattern of mobile nodes on streets. It can be useful in modeling movement in an urban area. The scenario is composed of a number of horizontal and vertical streets.
- Random Waypoint (RW): The Random Waypoint model is simple and the most commonly used mobility model in research community. At every instant, a node randomly chooses a destination and moves towards it with a velocity chosen randomly. E.g.: People wander around a campus, airport, museum or exhibition.

We have designed a new set of simulations under RW, MG and RPGM, to compare the operation of a-MMDSR with DSR in terms of packet losses. Two

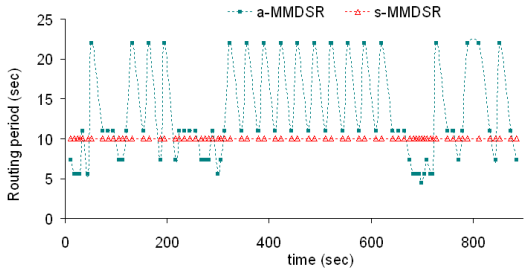


Figure 13: Evolution of $T_{routing}$ for a-MMDSR and s-MMDSR.

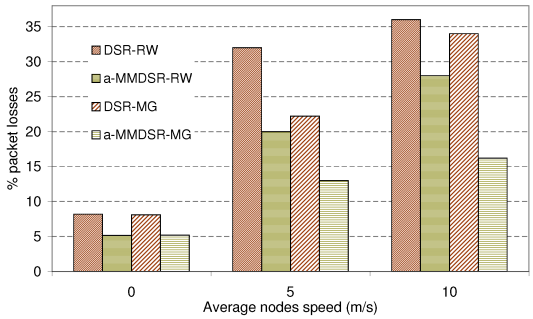


Figure 14: DSR *vs.* a-MMDSR under RW & MG mobility models (big scenario).

scenarios have been considered, a *big scenario* under RW or MG, and a *small scenario* under RW or RPGM. The simulation settings are the ones in Table 3, except for these values: $N$=100 nodes, $W$=500m, $R$=120m (*big scenario*) and $N$=20 nodes, $W$=250m, $R$=50m (*small scenario*).

Figs. 14 and 15 show the percentage of packet losses for DSR compared to a-MMDSR for each mobility model. In general, a-MMDSR outperforms DSR in both scenarios for the three mobility models. Also, it has been observed that DSR and a-MMDSR achieve fewer losses with RPGM compared to RW. This is because with similar relative speed between RW and RPGM, high degree of spatial dependence for RPGM means higher link duration and correspondingly higher path duration, which results in lower losses. In addition, MG presents more restricted movement of the nodes in the network compared to RW, which leads to slightly fewer broken links and subsequently lower losses. For the RW and MG mobility models in the big scenario, the packet losses of a-MMDSR compared to DSR plunges in average about 32% and 42%, respectively. For the RW and RPGM mobility models in the small scenario, the packet losses of a-MMDSR compared to DSR drops in average about 17% and 42%, respectively.

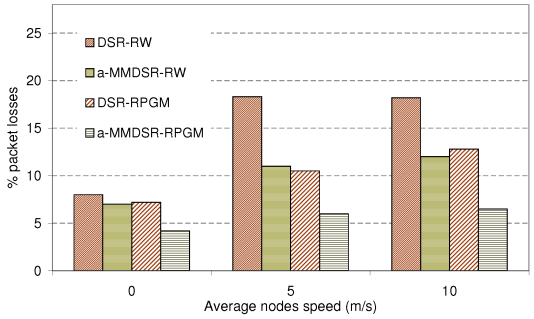


Figure 15: DSR *vs.* a-MMDSR under RW & RPGM (small scenario).

# 8 Conclusions and future work

In this article we have presented a QoS-aware self-configured adaptive framework to provide video-streaming services over MANETs (Mobile Ad Hoc NETworks). The proposal is named a-MMDSR (adaptive-Multipath Multimedia Dynamic Source Routing).

The main advantages of our approach arise from the fact that the framework is able to self-configure according to the dynamics of the environment. The routing algorithm periodically updates a set of paths, classifies them according to a set of metrics and arranges a multipath forwarding scheme. The benefits have been fulfilled by making parameters involved in the classification of the paths vary as a function of the network state, which is highly dynamic in MANET scenarios. Moreover, we have developed an analytical model to compute the path error probabilities, so that source nodes can proactively estimate the proper routing period in the set up moment of the multipath routing scheme. This way, the source knows the proper moment to refresh the multipath scheme before an actual breakage takes place in the path carrying the most important video frames. The proposal works in a different way under highly dynamic states than under more static situations, seeking to decrease the probability of having broken links and to improve the service performance, while using lower signaling overhead. Simulation results have shown that the proposal outperforms both the static version (s-MMDSR) and the plain DSR (Dynamic Source Routing).

Following with the design of a self-configured framework, we are currently developing a cross-layer algorithm based on reinforcement learning [21] techniques to make the network self-adjust its configuration parameters throughout time. The adaptation is based on the service experience of the user, and the system self-adapts seeking to provide the best performance. As future lines for this work, it would be interesting to implement all the system in a real testbed in order to have a closer idea of the system performance in a real scenario. Furthermore, it would also be useful to evaluate the effect of having heterogeneous networks. In this way, we would be able to evaluate our self-configuring framework in a real testbed when mobile nodes require a video-streaming service from the wired infrastructure.

# 9 Appendix. A numerical example

This appendix illustrates the approach by analyzing a particular case of study to evaluate the probability that a path breaks computed with formula (13). Let us consider these values for a specific network scenario: $N$=50 nodes, $W$=500m per side of squared area, $R$=120m of transmission range, speed deviation of the nodes $\sigma = 0$. The average speed of the nodes $\mu$ is a parameter that varies from 1m/s to 10m/s. In the following, we calculate all the parameters involved in equation (13).

We need to compute the maximum number of hops in a path from equation (14), which is $H$=5 hops in this example:

$$H = min(49, \lfloor \frac{\sqrt{2} \cdot 500}{120} \rfloor) = 5 \tag{19}$$

Next, we estimate the *Link Error Probability* (LEP) through time using equation (12).

$$LEP(R, \mu, t) \simeq 1 - e^{-\frac{\mu \cdot t}{R}} = 1 - e^{-\frac{\mu \cdot t}{120}}, 0 \leq t \leq 120/\mu \tag{20}$$

Finally, we complete expression (13) to estimate the probability that an existing path breaks, i.e. the PEP. Substituting (20) in equation (13) we have:

$$PEP(N, W, R, \mu, t) = 1 - \sum_{i=1}^{H} \left(1 - \left(1 - e^{-\frac{\mu \cdot t}{R}}\right)\right)^i \cdot \int_{(i-1)R}^{i \cdot R} pdf_{2n}(r) \cdot dr = \tag{21}$$

$$= \{N = 50 nodes, W = 500m, R = 120m, \mu, t\} =$$

$$= 1 - \sum_{i=1}^{5} \left(1 - \left(1 - e^{-\frac{\mu \cdot t}{120}}\right)\right)^i \cdot \int_{(i-1)120}^{i \cdot 120} pdf_{2n}(r) \cdot dr, 0 \leq t \leq 120/\mu$$

After calculating the integrals (16) and (17) and putting them into equation (21), we have:

$$PEP(N = 50 nodes, W = 500m, R = 120m, \mu, t) =$$

$$= 1 - (e^{-\frac{\mu \cdot t}{120}} \cdot 0.1457 + + e^{-\frac{\mu \cdot t}{60}} \cdot 0.3097 + e^{-\frac{\mu \cdot t}{40}} \cdot 0.3122 +$$

$$+ e^{-\frac{\mu \cdot t}{30}} \cdot 0.1930 + e^{-\frac{\mu \cdot t}{24}} 0.03781), 0 \leq t \leq 120/\mu \tag{22}$$

Figure 16 depicts the evolution of the path error probability over time which is expressed in eq. (22), for different average speeds of the nodes, from $\mu$=2m/s (7.2Km/h) to $\mu$=10m/s (36Km/h). As expected, the higher the speed of the nodes, the sooner the path would reach a higher probability of getting broken.

Next, we derive the proper period time of the routing algorithm ($T_{routing}$) using equation (18). For instance, if we set a maximum PEP of 60%, i.e. $P_{threshold} = 0.6$, according to equation (22) the source node should start looking for an alternative path for the routing mechanism before $T_{routing} \approx 11$ sec. for $\mu = 4$ m/s. This is pointed out in Fig. 16 with arrows. This way, our analytical model provides a tool with which a source node decides the optimum period of

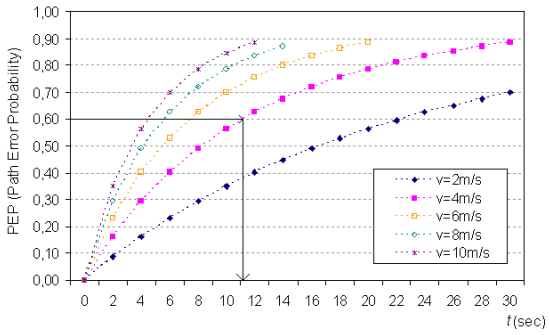


Figure 16: PEP throughout time with $\mu$ as a parameter ($R = 120$m, $W = 500$m, $H = 5$ hops, $\mu = 2, 4, 6, 8, 10$ m/s).

the routing algorithm to find an alternative path once the PEP of the current one exceeds a threshold.

Notice that $T_{routing}$ depends on the networks features and on the time, i.e. $T_{routing}(N, W, R, \mu, t)$. Therefore, our routing scheme works dynamically through time and adaptively to the network characteristics. In addition, integrals (16) and (17) are needed. However, since the expression to estimate PEP, i.e. eq. (21), depends only on the scenario settings (i.e. *N, W, R,* $\mu$) the final close and simple resulting expression can be computed at the beginning of the service. This way, $T_{routing}$ just depends on $t$.

We obtain $T_{routing}$ analytically from equation (22) just deriving $t = T_{routing}$ when PEP equals $PEP_{threshold}$. To solve the nonlinear equation (23), we apply numerical analysis such as the simple bisection method [16].

$$PEP_{threshold} = 0.6 = 1 - e^{-\frac{\mu \cdot T_{routing}}{120}} \cdot 0.1457 - e^{-\frac{\mu \cdot T_{routing}}{60}} \cdot 0.3097 - \\ -e^{-\frac{\mu \cdot T_{routing}}{40}} \cdot 0.3122 - e^{-\frac{\mu \cdot T_{routing}}{30}} \cdot 0.1930 + e^{-\frac{\mu \cdot T_{routing}}{24}} \cdot 0.03781 \qquad (23)$$

We have derived $T_{routing}$ for several speeds of the nodes and they are shown in Table 6. These moments are graphically depicted in Fig. 16. For example, for an average speed of de nodes of $\mu = 4$ m/s and a maximum PEP of 60%, the routing scheme should look for an alternative path after $T_{routing} = 11.19$ sec.. Table 6 shows that for higher nodes' speeds the $T_{routing}$ is lower (e.g. $T_{routing}$=4.51 sec for $\mu$=10 m/s), which is suitable since routes will probably break more frequently under such high mobility. On the contrary, for lower mobility the $T_{routing}$ self-adjusts to a higher value (e.g. $T_{routing}$=44 sec for $\mu$=1 m/s), which is suitable since routes are more stable and will probably remain longer. This adaptive computation of $T_{routing}$ leads to an efficient use of the resources, as the average routing overhead is reduced considerably (see Table 5) compared to a static assignation of $T_{routing}$.

Table 6 also shows the average path error probabilities obtained with and without the dynamic routing scheme for several speeds of the nodes, $\mu$. We can see that the average PEP with the dynamic routing scheme is considerably lower than without this scheme. Also, it is noticeable that whereas in the static

version PEP grows with $\mu$, it remains stable using the dynamic mechanism, as it is able to adapt to the speed. For instance, for $\mu = 6$ m/s, PEP decreases from 70% (without our scheme) to 33% (including our scheme). The routing protocol has to switch to an alternative path every 7.44 sec. Notice that without the dynamic routing scheme, the path would break around 20 sec. with 90% of probability (see Fig. 16 for $\mu = 6$ m/s). It may seem that it would be preferable to wait longer to switch to alternative paths in order to reduce the routing signaling overhead. However, a path break would produce a severe damage in the subjective experience of the user. We have measured a 6% percentage of signaling overhead for $PEP_{threshold}$=60% and 5% for $PEP_{threshold}$=90%. There is a trade-off between QoS and signaling overhead, which seems to be well-balanced for a $PEP_{threshold}$ around 60%.

Table 6: Routing period of the algorithm ($T_{routing}$) for different average speeds of the nodes. PEP with and without the dynamic scheme.

| Average Path Error Probability ($\overline{PEP}$) (%) | | | |
|---|---|---|---|
| $\mu$ (m/s) | $\overline{PEP}$ without $T_{routing}$ (%) | $\overline{PEP}$ with $T_{routing}$ (%) | $T_{routing}$ (s) |
| 1 | 38 | 31 | 44 |
| 2 | 56 | 32 | 22.32 |
| 3 | 67 | 31 | 14 |
| **4** | 73 | 30 | **11.19** |
| 5 | 77 | 30 | 8 |
| **6** | **70** | **33** | **7.44** |
| 8 | 76 | 35 | 5.56 |
| 10 | 79 | 29 | 4.51 |

## 10 Acknowledgements

This research article is supported by the Spanish projects SECONNET (CICYT-TSI2005-07293-C02-01), ITACA (CICYT TSI2007-65393-C02-02) and UPC Research grant. The authors would like to thank the anonymous reviewers for their careful reading and insightful comments that have helped in improving the presentation of this paper.

## References

[1] Azzedine Boukerche, "Algorithms and Protocols for Wireless Mobile Ad Hoc Networks", Wiley-IEEE Press, 2008.

[2] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod, "Cross-layer Design of Ad Hoc Networks for Real-Time Video Streaming", *IEEE Wireless Communications*, Vol. 12, Issue 4, pp. 99-102, 2005.

[3] V. Carrascal Frías, G. Díaz Delgado, M. Aguilar Igartua, "Multipath Routing with Layered Coded Video to Provide QoS for Video-streaming applications over MANETS", *14th IEEE International Conference on Communication Networks* (ICON), 2006.

[4] A. B. McDonald, T. Znati, "A mobility-based framework for adaptive clustering in wireless ad-hoc networks", *IEEE Journal on Selected Areas in Communications*, pp.1466-1487, 1999.

[5] A. Treviño Cabrera, J. García de la Nava, E. Casilari, F.J. González Cañete, "An Analytical Model to Estimate Path Duration in MANETs", *The 9-th ACM/IEEE Intern. Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (MSWiM06), Spain, pp. 183-186, 2006.

[6] E. Hua, Z. Haas, "Path selection algorithms in homogeneous mobile ad hoc networks", *Int. Conf. Comm. & Mobile Computing*, pp. 275-280, 2006.

[7] IEEE 802.11e standard with Quality of Service enhancements, *http://standards.ieee.org/getieee802/download/802.11e-2005.pdf*

[8] P.N. Tudor, "MPEG-2 video compression", *Electronics & Communication Engineering Journal*, Vol. 7, No. 6, pp. 257- 264, 1995.

[9] RFC 4728: The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks, February 2007, *ftp://ftp.rfc-editor.org/in-notes/rfc4728.txt*

[10] V. Loscri, F. Rango, S. Marano, "Performance evaluation of on-demand multipath distance vector routing protocol over two MAC layers in mobile ad hoc networks", *Int. Symp. Wireless Comm. Systems*, pp. 413-417, 2004.

[11] V. Carrascal, G. Diaz, A. Zavala, M. Aguilar, "Dynamic cross-layer framework to provide QoS for video-streaming services over Ad Hoc networks", *ACM QShine*, Hong Kong, 2008.

[12] Víctor Carrascal Frías, "Contribution to provide QoS over Mobile Ad Hoc Networks for Video-Streaming Services based on Adaptive Cross-Layer Architecture", PhD Disertation, 2nd March 2009, Advisor: Mónica Aguilar Igartua, Dept. of Telematic Engineering, Technical University of Catalonia, Barcelona, Spain. Available in *http://sertel.upc.es/tesis.php*. NS-2 contributed code available in *http://globus.upc.es/ vcarrascal/ns2/*.

[13] L. E. Miller, "Distribution of link distances in a wireless network", *Journal of research of the National Institute of Standards and Technology*, Vol. 106, No. 2, pp. 401-412, 2001.

[14] Y.-T. Wu, W. Liao, C.-L. Tsao, "Epoch distance of the random waypoint model in Mobile Ad Hoc networks", *IFIP International Federation for Information Processing*, LNCS 3731, pp. 518-524, 2005.

[15] D. Hong, S. Rappapport, "Traffic models & performance analysis for cellular mobile radio telephone systems with prioritized and non prioritized handoff procedures", *IEEE Trans. Veh. Tech.*, 35(3), pp. 77-92, 1986.

[16] Bisection Method, University of South Florida, *http://numericalmethods.eng.usf.edu/topics/bisection_method.html*

[17] The Network Simulator *ns-2, http://nsnam.isi.edu/nsnam/*

[18] BonnMotion, Mobility scenario generation, *http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion/*, 2005

[19] ITU-T, "Mean Opinion Score (MOS), methods for objective and subjective assessment of quality", Recommendation ITU-T P.801, *International Telecommunication Union*, 1996.

[20] L. N. Cai, D. Chiu, M. McCutcheon, M. Robert Ito, G. W. Neufeld. "Transport of MPEG-2 video in a routed IP network", *Lecture Notes in Computer Science*, DMS, pp. 5973, 1999.

[21] R. S. Sutton, A. G. Bartovv, "Reinforcement Learning: An Introduction", *The MIT Press*, 1998.

**Mónica Aguilar Igartua** is an associate professor at the Tech. Univ. of Catalonia (UPC), member of the Telematics Services Group at the dept. of Telematics Engineering of the UPC (Barcelona, Spain). She received her MSc and PhD degrees in Telecommunication Engineering from the UPC in 1995 and 2000 respectively. Her research activity includes modeling and evaluation of multimedia services over heterogeneous networks, design of dynamic self-configured QoS-aware frameworks to provide video-streaming services over multihop wireless networks, such as MANETs (Mobile Ad hoc Networks) and VANETs (Vehicular Ad hoc Networks).

**Víctor Carrascal Frías** received his MSc and PhD degrees in Telecommunication Engineering from the Technical University of Catalonia UPC (Barcelona, Spain) in 2003 and 2009, respectively. His research interests include the provision of video-streaming services over IP networks. He has developed ns-2 implementations of QoS-aware multipath routing protocols for Mobile Ad hoc Networks. He is currently working at GTD in Barcelona as networking engineer.